1. **執行環境**

* Jupyter Notebook

1. **程式語言&版本**

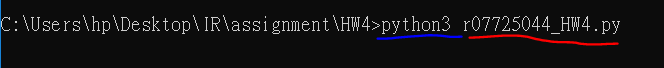
* 語言：Python
* 版本：3.6.6

1. **執行方式**

打開cmd，cd到檔案位置



再輸入以下指令，即可執行並顯示output：



藍色部分填入python.exe及其路徑，或是如上圖輸入在python開啟的指令，python3是我在環境變數下設定python 3.6.6版本的指令

（在環境變數先加入python.exe的path，再更改檔名為python3.exe,即可直接啟動）

紅色填入檔名 HW4\_r07725044.py

得到output：

在當前目錄下會產出名爲8.txt、13.txt、20.txt的檔案，是分群後的結果

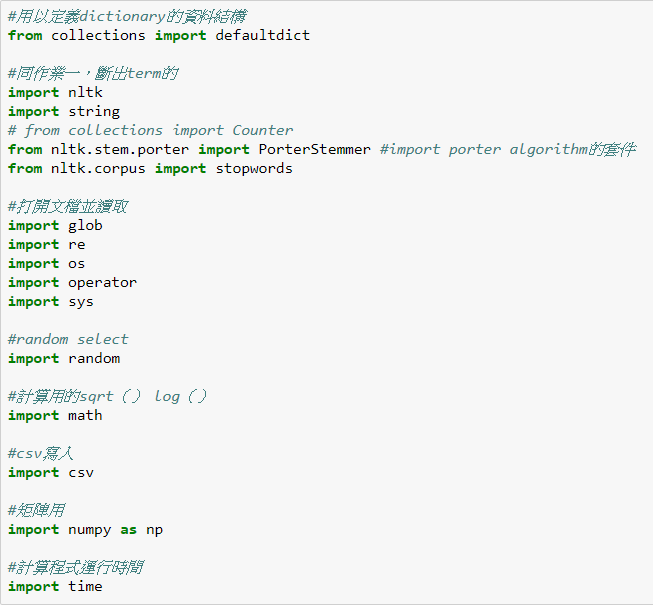
此外還會產生一個名爲matrix.dat的檔案，是記錄文件之間相似度的matrix

1. 作業處理邏輯説明

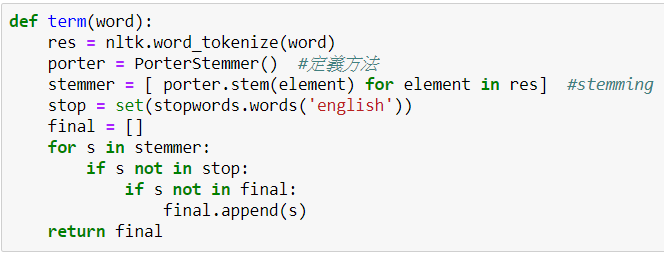
***1. 前處理***

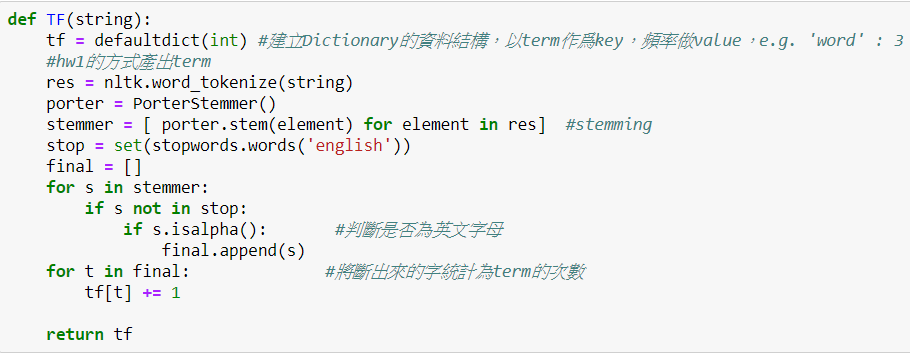
* + **Import 套件**

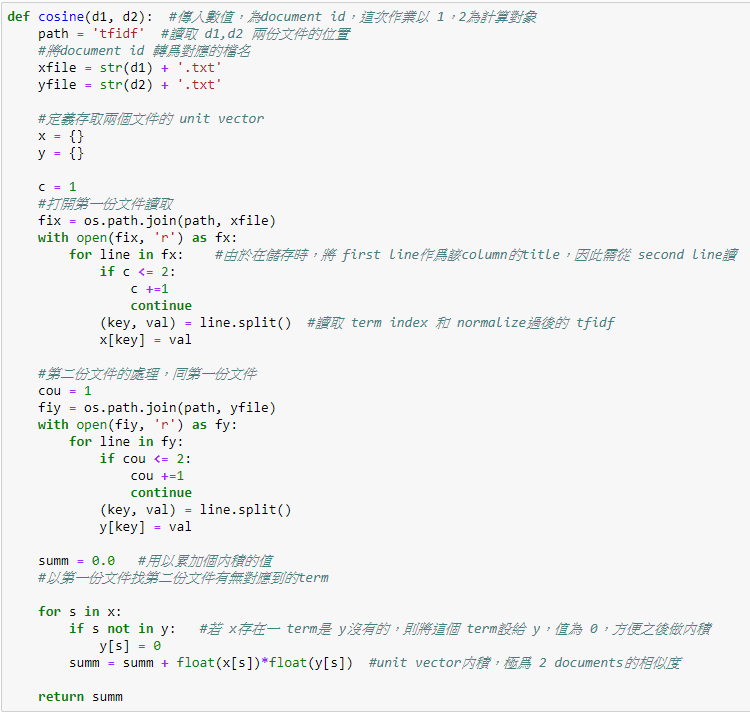
將所有要用的套件import ，用途如註解



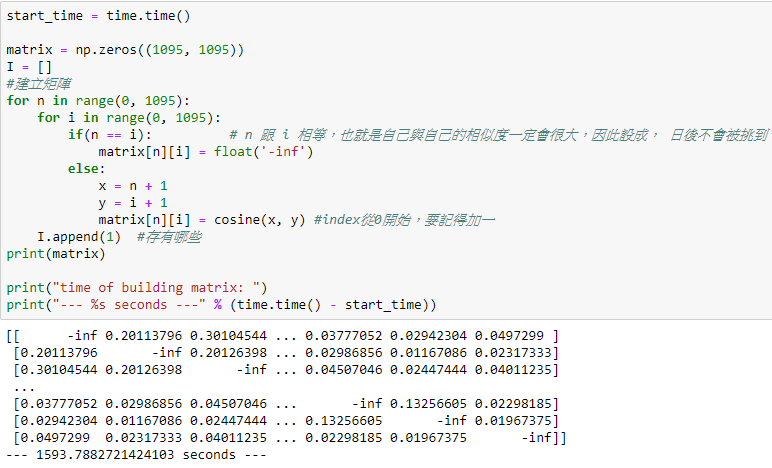
* + **加入前幾次作業的function**

將document轉換成term的function

計算term frequency的function ：

計算各文件之間的相似度：

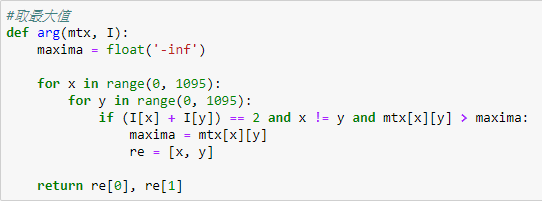
* + **建立各文件之間相似度的matrix**

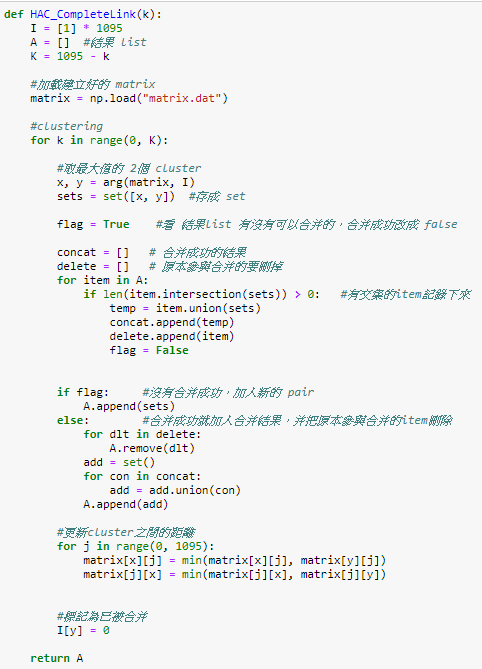
運用cosine similarity的方法，建立文件之間相似度關係的matrix，值為0~1，若是文件對自己本身的關係，則設爲負無限大，如此一來不會在argmax的時候取到



1. ***Clustering***
   * ***Hierachical Clustering + Complete Link***

Argmax的function：

2個cluster必須都是available的，才能被取值

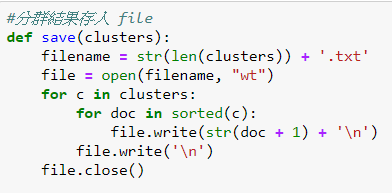
Clustering

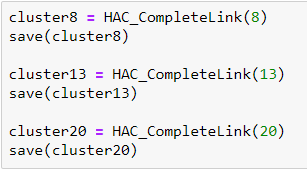
取出最大值的兩個對應的cluster，把編號較大的合并進小的，且用set的資料結構，將包含2個cluster其中的任意一個的cluster全部合并，最後更新cluster之間的相似度關係，使用complete link所以取最小相似度，再更新被合并的cluster為not available

* + **Call function and save result**

儲存至file的function：

因爲前面的資料結構是以0為開始的，所以所有編號要加1才是正確的文件編號，每個cluster中間隔1行

呼叫function，分別分出 8群、13群、20群：



最後當前目錄會產生出3個以群數為命名的檔案，就是分群結果

